Software Communications Architecture Specification

APPENDIX E. SERVICE DEFINITION DESCRIPTION

Revision Summary

1.0	release for prototype implementation and validation

Table of Contents

APPENDIX E SERVICE DEFINITION DESCRIPTION

E.1 DESCRIPTION / PURPOSE.

A Service Definition specifies the interfaces (requests, responses, indications, confirmations, and acknowledgements), behavior, state information, and exceptions that define a particular service.

The Service Definition can be used to implement a service or define a basic building block. A basic building block is an abstract Service Definition that must be instantiated with concrete types to be realized. Building blocks are used to define abstract services to foster reuse and commonality between different implemented Service Definitions.

A Service Definition is used as the basis for implementing services (Service Providers) which can be used by Service Users without regard to the particular implementation of the service.

Note that text in *italics* is included as example material. Wording within <> represents information that shall be supplied by the developer.

E.2 REUSE.

Service Definition reuse can be accomplished via either inheritance or copying. When the complete functionality of an existing Service Definition is desired then that Service Definition shall be inherited. If only a portion of the existing Service Definition is desired, that portion of the Service Definition required can be copied from the existing Service Definition.

E.3 PREPARATION INSTRUCTIONS.

E.3.1 Format.

The format of a Service Definition shall follow the structure of the following sections. The third level of heading in this document shall be the first level of heading for a Service Definition. Each Service Definition shall contain a cover page and a table of contents. At a minimum, the cover page shall include a title, UUID, version number, and point of contact (name, organization, address).

E.3.2 Content.

E.3.2.1 Introduction.

E.3.2.1.1 Overview.

This section contains an overview of the service. Included in this overview shall be a list of supported protocols, algorithms, and waveforms.

E.3.2.1.2 Service Layer Description.

The role of the service layer shall be described here in high level terms.

E.3.2.1.3 Modes of Service.

This section shall identify the different modes of the service, if any. Included with the identified modes shall be a description of use. In general, developers should try to avoid defining modes but if modes are defined do no use modes that cause invalid operations.

{TBR}

E.3.2.1.4 Service States.
This section shall identify the states for the services.

E.3.2.1.5 Referenced Documents.
Documents referenced by a Service Definition shall be listed here. Included here would be inherited Service Definitions with UUIDs and any protocol and waveform standards that describe or define the service.

E.3.2.2 UUID.

This section shall provide the UUID for the Service Definition.

E.3.2.3 Services.

This section shall provide a brief overview of services and a table of service groups, services, and the primitives that support those services. An example is shown in section E.4.1.

E.3.2.3.x <Service Group or Service>
This section shall identify a service group or service and provide a description of the service. If the service can be more clearly specified by further dividing into sub-services, the sub-services shall be specified in subparagraphs. Each identified service shall be accompanied by a brief explanation. A time sequence diagram shall accompany the lowest level of service identified. Examples of time sequence diagrams include the type defined in ISO/IEC 10731 Conventions for the Definition of OSI Services, Annex D Alternative and Additional Time Sequence Diagrams for Two-party Communications Section 2 or a UML sequence diagram. An example is shown in section E.4.2

E.3.2.4 Service Primitives.

E.3.2.4.x <Service>
The primitives shall be grouped according to the services defined in section 2. There may be reasons for additional groupings that the service author may add for clarity. This section should provide UML class diagram(s) that represent the interface(s) for the service to provide an overall view of the service.

Inheritance relationships from Service Definitions not defined in this document shall be described and should be shown in the UML class diagrams. If primitives inherited by this Service Definition remain unchanged by the context of the inheriting service, then section 3.2.3.x.y and its subparagraphs may be omitted for those primitives and a list of the inherited primitives and references to their sources may be provided instead. Each primitive shall start a new page in the Service Definition.

The implied get and set operations for attributes shall be explicitly defined as primitives.

E.3.2.4.x.y <Primitive Name>
This section shall contain a brief description of the primitive.

E.3.2.4.x.y.1 Synopsis.
This section shall contain the syntax of the primitive. Additional explanatory text may be provided as well.

E.3.2.4.x.y.2 Parameters.
This section shall identify and describe the parameters for the primitive. If a structure or structures are part of the parameter set, then each field shall be enumerated and described.

E.3.2.4.x.y.3 State.
This section shall identify the state(s) for which the primitive is valid.

E.3.2.4.x.y.4 New State.
This section shall identify the new state that results from the execution of the primitive, if any.

E.3.2.4.x.y.5 Response.
This section shall identify the response primitive, if any.

E.3.2.4.x.y.6 Originator.
This section shall identify the originator of the primitive, the Service User or the Service Provider.

E.3.2.4.x.y.7 Errors/Exceptions.
Any errors or exceptions that can occur as a result of the execution of the primitive shall be identified and described.

{It is planned to give an example of the primitive in future releases.}

E.3.2.5 Allowable Sequence of Service Primitives.

This section shall describe the allowable sequence of service primitives that may be issued across the interface. The following state information shall be supplied: state name, transition primitive, expected outputs, and exceptions.

{It is planned to give examples in future releases.}

If there are no defined sequences of primitives, this appendix shall be left blank.

E.3.2.A Precedence of Service Primitives.

This appendix shall present a summary of the precedence of service primitives as they are queued by the Service Provider and/or Service User. The precedence shall be expressed in tabular form. If there is no precedence of primitives, this appendix shall so state.

E.3.2.B Service User Guidelines.

This appendix shall summarize the guidelines for implementing a Service User that will be independent of the implementation of the Service Provider.

E.3.2.C Service Provider-Specific Information.

This appendix shall identify the information to be documented for each service provider implementation.

E.3.2.D IDL.

This appendix shall include the complete IDL listing for the Service Definition. This includes the IDL for any inherited Service Definition(s). New IDL shall be differentiated from the inherited IDL in some fashion (e.g. different font).

{TBR.}

E.3.2.E UML (Optional).

This appendix, if provided, shall include the UML class and component diagrams for the Service Definition. The purpose for including these diagrams is to show the relationship between all the elements that make up the service.

E.4 EXAMPLES

E.4.1 Example Service Table.

Service Group	*Service*	*Primitives*
Connection Establishment	***Connection Establishment***	***CONNECT_REQ, CONNECT_IND, CONNECT_RES, CONNECT_CON, DISCONNECT_REQ, DISCONNECT_IND***
Data Transfer	***Data Transfer***	***DATA_REQ, DATA_IND***
	Expedited Data Transfer	***X_DATA_REQ, X_DATA_IND***

{This example will be updated with work developed for building blocks.}

E.4.2 Example Service Groups and Services.

The following example shows the management service separated into constituent services in subparagraphs with the Attach Service filled in.

{This example will be updated with work developed for building blocks.}

3.1 Management Services

The local management services apply to the connection, connectionless and acknowledged connectionless modes of transmission. These services, which fall outside the scope of standards specifications, define the method for initializing a stream that is connected to a DLS provider. DLS provider information reporting services are also supported by the local management facilities.

3.1.1 Information Reporting Service

3.1.2 Attach Service

The attach service assigns a physical point of attachment (PPA) to a stream. This service is required for style 2 DLS providers (see Physical Attachment Identification) to specify the physical medium over which communication will occur. The DLS provider indicates success with an OK_ACK; failure with a ERROR_ACK. The normal message sequence is illustrated in the following figure.

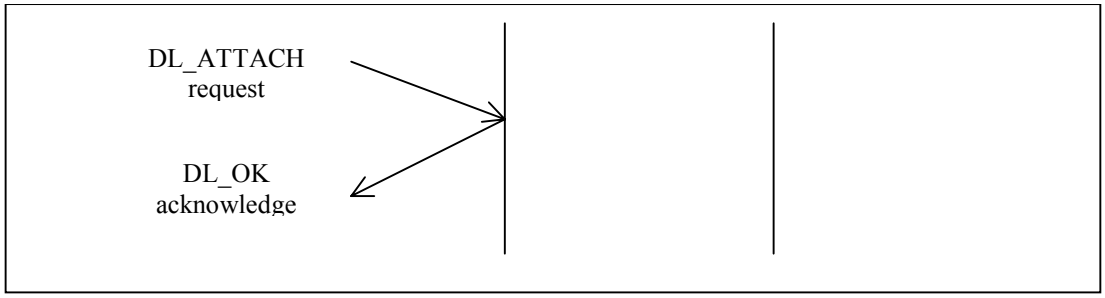

Figure: Attaching a Stream to a Physical Line

A PPA may be disassociated with a stream using the DETACH_REQ. The normal message sequence is illustrated in the following figure.

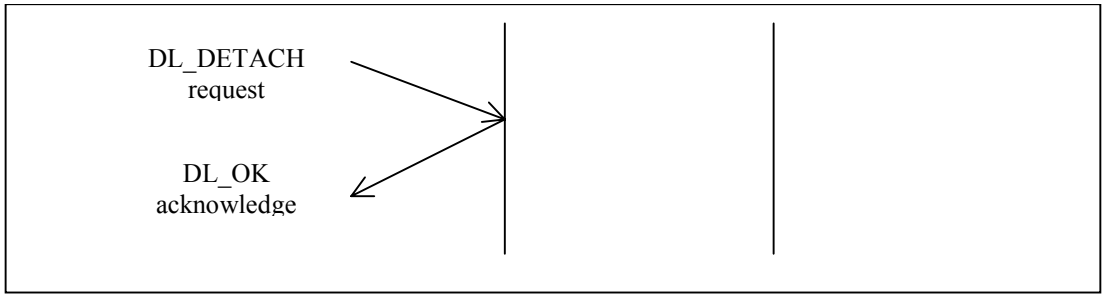

Figure: Detaching a Stream from a Physical Line

3.1.3 Bind Service